



# SEQUENCE PLANNING OF TASKS WITH IDENTICAL WORK PROCESSES FOR ACHIEVING THE SHORTEST POSSIBLE TIME OF REALIZATION

T. TÖRÖK and Z. VATTAI

Department of Building Management and Organisation  
Technical University Budapest, H-1521

Received: March 1988

Presented by Assoc. Prof. Kürti

## Abstract

It would be very difficult to list, and even to survey, where and when in practice the determination of optimum sequence can play a role. Operation research has already elaborated some (more or less specific) procedures for determining optimum sequences. In many cases, the production system itself does not allow the managers and organizers of production to adjust the optimum operation of their units to their inner nature (subcontractors, outside workers, transporters, energy, etc.). Even where a possibility would arise for a self-supporting, inner organization of work, it should be studied whether the sequence planning is a real or only an imaginary problem. That is, how large an effect the production of products in different sequences may have on the parameters of production (time, money, energy, etc.). For cases where the determination of optimum sequences is a real demand, the following can be said: In performing sequence planning it is often expedient to estimate the optimum by some method, and the study itself may be a partial enumeration procedure by utilizing the large capacity of up to date computers. During this, such sequence (sequences) should be searched for, the target value of which is the supposed optimum, or the value less worse than that. (Though this can also require significant machine time, it brings the solution of the problem closer.) The procedure shown is constructed on the basis of this principle and by its aid we succeeded in solving sequence planning tasks of such dimension which could not be even tried to be solved until now due to the large number of possible solutions.

## 1. Introduction

In sequence planning the problem to be solved is how at certain bottlenecks\* of the production (machine, specialized group of workers (brigade), plant, subcontractor, etc.) the jobs with the narrowest cross-section of realization time should be sequenced to achieve highest efficiency in the utilization of bottlenecks from the point of view of some objective.

Due to limited capacities, fixed technological sequence and different partial processing requirements of products, the production sequence of individual products may have a strong influence on the necessary total production time. The minimum total production time may differ by several hundred per cents (!) depending on the sequence of the production of individual products. (By total production the total of products produced in one plan or production period is meant.) At the Department of Building Management and

\* See in more detail in "Operation Research" by G. Jándy, Tankönyvkiadó, Budapest.

Organization of the Technical University the research project supported financially by the state: "The PC-aided development of system analysis and operation research in the service of architecture and transportation" serves besides the development of algorithms, computer programs helping gradual and post-gradual education, also the elaboration of PC-aided sequence planning tasks which are very important in construction work.

What in the industry the workpiece to be processed is, is in construction the building itself. The architectural productive unit (main firm, brigade, machine, etc.) is analogous to the machine. The only difference from the viewpoint of organization is that whereas the workpiece can be processed only by one machine simultaneously, on buildings several working groups can work at the same time. This difference becomes also indistinct, if instead of individual workpieces, series of products are considered, or if in construction, subsequent work processes can only be started after finishing the preceding ones.

It is characteristic for both fields that the products of identical nature are produced by several similar or identical procedures at a given production sector by units specified for the partial work processes, mostly in a given technological sequence.

Thus the task to be solved in construction industry can be formulated as follows: the construction sequence of a given number of buildings ( $i = 1 \dots \dots n$ ) should be determined so that the construction time for the total of buildings be a minimum. In the construction of every individual building, a given number of work processes ( $j = 1 \dots \dots m$ ) should be carried out in a given sequence. The time requirement of individual work processes is known ( $d_{ij}$ ), as well as the time lapse between the individual work processes (the follow-up time in the construction of building  $i$  between work processes  $j$  and  $j + 1$  is designated by  $z_{ij}$ ). The follow-up time is the minimum period between the ends or starts of subsequent work processes, which is given in the practice of network planning by the dependence of critical approximation ( $CR_z$ ).

In the course of the solution of the above task it may also be required that certain work processes should proceed continuously through the buildings, e. g. in cases where the continuous work of individual brigades or machines is a basic requirement. Naturally, if the working groups carrying out individual processes are employed continuously, usually different sequences result in an optimum term of production, and this time is greater than or at most identical with that obtained if no continuous work is required.

## 2. Principles of the solution of sequence planning

In the mathematical description of sequence planning, a task of integer nature should be formulated in which the decision variable expresses if a given

building represents a given element of a given sequence or not. However, due to the size (the great number of conditions and variables), complexity and the requirement of being integer values of the mathematical model, the computer programming of the task is out of the question. In such cases, operation research suggests the application of one of the following three main procedures:

- enumeration algorithms, i.e. the enumeration of all possible solutions (total enumeration),
- “branch and bound”, called also partial enumeration,
- heuristic procedures.

The difficulty in the solution of sequence planning is that the number of possible sequences increases factorially with the increase in the number of elements to be sequenced. For example, the possible number of sequences for 4 elements is 24, for 5 elements 120, for 6 elements 720, for 7 elements 5040 and for 20 elements it is already  $2.43 \times 10^{18}$ . This is such a great number which is imperceptible for human brain; it could be visualized by imagining a computer capable of calculating 1 million different schedules in a second. This computer would need 77 thousand years for calculating all the possible schedules. Such a computer does not even exist as yet, and the number of buildings to be constructed simultaneously may even exceed 20. Thus in practice, the complete computer solution of a sequence problem cannot be realized, the sequence is rather determined by considering other aspects. The solution of problems by computer programming is limited for 5—6 buildings. In the procedure shown in this work, we wanted to loosen somewhat on this very strict restriction, and to provide a PC-aided solution of sequence planning for 15—20 buildings.

In the elaboration of the solution procedure, the following problems have been encountered:

- total enumeration would require impossibly long times,
- the partial enumeration procedures known from the literature would necessitate an extremely large storage capacity,
- there are some heuristic procedures for sequence planning, such as e. g. the Johnson rule which provides help in sequencing several buildings through 2 bottlenecks, but they are not applicable for our task.

Knowing the above problems, we decided to elaborate a computer procedure surely leading to the optimum sequence or sequences — thus a priori no heuristic procedures were considered — and studying the possible sequences in a lexicographic sequence (thus the large storage capacity needed for the branch and bound procedure is not required), and at the same time making a partial enumeration decreasing thereby the running time, i. e. in the search for the optimum sequence applying a variable limiting value it excludes certain partial groups of possible solutions.

The combinatorical enumeration of the total of possible solutions for sequence planning can be visualized by the so-called enumeration tree, which

is a graph representing all sequences so that one and only one pathway corresponds to a given sequence, and only once a node is presented on each pathway. A given level of the enumeration tree of a sequence planning shows which element of the sequence is concerned, whereas sub-trees mean the set of sequences originating from a given point of the tree.

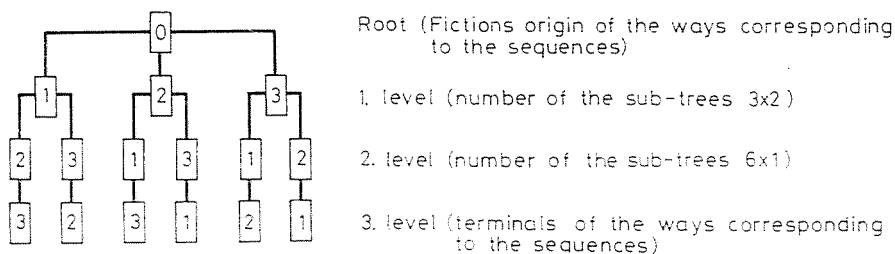


Fig. 1. Structure of the enumeration tree of corresponding to the sequences of 3 possible elements

From the enumeration tree shown in Fig. 1 it is apparent that only one pathway corresponds to each individual sequence lexicographically increasing, since if the code numbers of the individual pathways are read together as a number, these numbers increase when passing from the left to the right:

$$123 < 132 < 213 < 231 < 312 < 321$$

During the elaboration of the solution for our sequence planning task, following "ordering principles" — partially of a computerprogramming nature — have been formulated:

- a. In the search for an optimum, a so-called first approximating value should be looked for,
  - b. the program should not in every case provide the whole enumeration tree, but it should be able to do that, if needed,
  - c. the algorithm for the calculation should be the shortest possible, and should contain the simplest possible operation for shortening the running time,
  - d. only sequences leading to an ever better or at least identical target value should be considered,
  - e. sequences corresponding to the actual limit value should be displayed immediately,
  - f. only a minimum number of intermediate (calculated) data should be stored,
  - g. interactivity should be ensured at the highest possible level.
- a. In searching for an optimum, a first approximation should be looked for as a filter, based on the estimation of the optimum, moreover, this is even necessary for the minimizing of the computing time.

b. If necessary, the program should be able to provide the whole enumeration tree in order to reveal all the possible optimums. At the same time, it should be ensured that in the course of search for the optimum, the evaluating algorithm should be capable of interrupting the study of the enumeration sub-tree at any given level of the tree and preceeding to the next sub-tree if the target value exceeds the actual limit. This serves the shortening of the enormously large running time of total enumeration.

The computer solution of the task can be deduced from an algorithm organized around an evaluating core embedded into a double loop, where the outer loop controls the cycle variable of the inner loop, and at the same time, the evaluating "core" embedded into the inner loop controls both the outer and inner loops at a given level of the tree. The inner loop ensures proceeding from the left to the right, whereas the outer loop ensures the "up and down" motion between the levels of the tree. It is basically important from the viewpoint of running time, at which level of the tree the inner core interrupts the study of the given sub-tree. Namely, the lower the level at which this interruption occurs, the faster the time needed for the study of the enumeration tree increases. This shows the importance of a relatively good estimation of the optimum by the first approximation.

c. Since the filtering, evaluating algorithm operates at the "nodes" of the enumeration tree, its rate also influences the running time of the program. However, this influence is smaller than that of the above "interruption level".

d. In every problem solving procedure it is a basic requirement that it should be purposeful. This can only be achieved if the target values of the sequences studied become better and better, or at least not worse from the point of view of the optimization characteristic. Thus in the evaluating core, decision about an eventual interruption in the study of the sub-tree should be made, based always on the last and best target value.

Due to the nature of partial enumeration procedures, i. e. that they search for and not determine the optimum sequence or sequences, the rate of the solution is very data dependent.

The time needed is also strongly influenced by the extent to which the target value of the real optimum differs from that of the other sequences. If, namely, the optimum shows very markedly, the non-optimum nature of a given sequence manifests itself already at a relatively high level of the enumeration tree. However, if the optimum is not very definite, the filter interrupts the study of the sub-tree only at a lower level. This problem can be moderated by the realization of interactivity at a higher level.

According to experience, the optimum searching program can be quickened significantly, if only sequences leading to ever better target values are considered. However, no alternative optimums can be obtained this way, as the program would provide the first optimum found on the enumeration tree

— which is essentially alternative —, as the possibility for an evaluation according to other aspects would be excluded. It has already been mentioned that the rate of the evaluating routine has a smaller influence on the rate of the solution than the level of interruption. By utilizing this, it may be expedient to build several subordinated aspects into the evaluating core, when the routine can choose further from among the alternatives found according to an aspect on the basis of further subordinate aspects. However, there is no guarantee that the sequence thus chosen is also optimum for the subordinate aspects. Therefore the targets should be unambiguously ranked. The evaluation aspects of first, second and so on order could, of course, include several aspects if the “unified” evaluation function may be described as a combination of the functions of the composite target functions. By these solutions, the interruption levels can be pushed higher on the tree, and at the same time, the task approaches a kind of multifactorial optimization problem. However, it should not be left out of consideration that the procedure searches for an optimum sequence and does not determine it.

e. Immediate display of the appropriate sequences is needed due to the restricted storage capacity. First it should be considered here that in the majority of tasks more, eventually very many different optimum sequences exist, but their number is not known previously. Thus, if the enumeration of all, alternative optimum sequences is required, a data recorder of “unrestricted” capacity is needed. In practice, this would mean a paper tape of “infinite” length.

f. Due to a lexicographic passing through the tree — from the left to the right — a return to a sequence already studied is excluded. Consequently, it is enough to store the data concerning the nodes of the last sequence, which means a number of intermediate calculated data at most by one less than the number of elements to be sequenced. Thus the possibility arises for the simultaneous storage of several data concerning the sequences studied. This provides also the possibility for choosing the appropriate sequence according to more (unequivocally ranked) aspects.

g. In order to be of much help, computers should perform programs most elastically, most openly, i. e. they should be interactive. But not only during data input or data preparation — what is nowadays almost exclusively so — but also during run. In many cases it is required that the algorithm running could be interrupted at any arbitrary place and time, its parameters could be modified, stepped back, etc. This is especially necessary in enumeration studies, as the actual search can be formulated in a single routine which, under unfavourable conditions, would operate for a very long time. Therefore it is indispensable that the user could interrupt the study at any point of the tree, could have an insight into the actual parameters and the place of interruption. Based on this, he could make decisions concerning the further run of calcula-

tions, could let the search go further, could step on the tree forward or backward, modify the parameters, prescribe new studies, etc.

The above "services" can easily be realized and they decrease the rate of run only insignificantly, increasing at the same time remarkably the connection between machine and user, that is the efficiency of the procedure.

### 3. Procedure

In our task, the periods of work processes and the approximation relations determined by the technology are known for each building. The technological sequences of jobs are also known and identical for each building. The solution essentially means an ordering into an arhythmic belt so that the total construction time be a minimum.

One of the most important logical building stones of the procedure is the following recognition:

The total construction time consists of two, well separable parts:

*a*: the total processing time of the first work process (this is identical in every sequence),

*b*: the period between the finishing of the first and last work processes on the last building.

This latter can further be separated into two parts:

*b1*: The minimum period given by the starting data, i. e. a technologically possible minimum of period "b" (this can be determined previously relatively easily),

*b2*: the part of period "b" depending only on the sequence.

This is the period on the basis of which a limited branching can be carried out. It is usually 0—30% of the total processing time relative to the most favourable sequence, but may theoretically reach several hundred per cents (!).

The cardinal point of the solution is, how accurately the value of period "b1" can be determined, as the value of the filter used in limited branching can be determined from this period. The sum of periods "a" and "b1" can be utilized for estimating the optimum value.

Let us now see the principal scheme of the procedure:

#### *Step 1:*

On the basis of the time requirements of work processes and the critical approximation, the technologically possible minimum period of the construction of individual buildings is determined, and also the schedules of work processes belonging to it. Let us call them the "own" schedules of buildings. From these, the periods between work processes (at the beginning and at the

end) can be determined from building to building, and they may be described by 2 vectors of  $m-1$  dimension for each building. The important logical building stone of the procedure is, namely, that the length of period "b2" depends only on the relation of these values to each other. Namely, the more parallel is the "ending" vector of a preceding building to the "beginning" vector of the following one in the sequence, the simpler is for the work groups to pass from the preceding building to the following one, thus the total schedule becomes more "closed" (narrower), i. e. the total construction time becomes shorter.

### *Step 2:*

#### Estimation of the optimum value

As it has been mentioned above, this is the most delicate point of the procedure. Following theoretical algorithm is built into the computer program:

From the total processing time summarized by work processes, as work processes of fictive, constant intensity (linear), — considering the continuity requirements of work processes — we prepare the closest possible schedule in which the critical approximations of subsequent work processes are fixed at 0. Let us call it a fictive schedule. (By continuity requirement it is meant whether the interruptability or strict continuity of work processes is prescribed for the passing from one building to the other. This can be decided also for individual work processes.) In this fictive schedule, the ending and beginning relations of work processes can also be described by 2 vectors. After determining them, the computer chooses from among all possibilities the processing time of the schedule proved to be shortest by fitting it to the beginning of a building selected for the starting one and the ending vector of a different building selected as the ending one as the starting point of optimum estimation, i. e. as filter. This means  $n \times (n - 1)$  trials (where  $n$  is the number of buildings). The processing time of the optimum sequence will certainly be not shorter than this. With this value as filter, the search for the optimum may begin. The value of the starting estimation is displayed, thus it informs the user about the expectable value of the optimum. Before search, there is of course a possibility for modifying the filter, the importance of which will be clear in the next section.

### *Step 3:*

In the course of searching, the program combines the "own" schedules of buildings until the total processing time exceeds the value of the filter due to fittings. The algorithm for evaluating the fittings requires calculations with simple operations, but eventually multiply recursive ones. Its description exceeds the limits of this paper. During fitting, usually a loosening, spreading



of the already sequenced or to be sequenced schedules occurs (i.e. the follow-up times increase), ensuring thereby the starting technological and continuity requirements. The search starts with the selection of the last building, as periods "b" and "b2" can be thus observed continuously from the beginning. After that the one but last, then the preceding building is chosen and so on, thus fitting occurs backwards from the schedules of buildings for determining the sequence. During the study increasing in a lexicographic sequence, the first element thus means the selection of the last building in the sequence. Upon observing the extent to which the already sequenced schedules have to be spread, we can follow whether the processing time of sequences containing the already fitted chain (they end with it), exceeds the value of the filter or not. The program decides on the basis of this on the continuation of sequencing or on the modification of the sequenced chain, i.e. on the exclusion of a significant part of possible sequences from the further study. If the program does not find any sequence corresponding to the filter, it starts the search again after an automatic increase in the value of the filter. If the program does find a sequence corresponding to the filter, it proceeds further on 3 possible pathways according to the instruction of the user, in order to ensure a more elastic practical application:

a. It stops the search immediately after finding the first appropriate sequence. This can be useful in the case of very large tasks, when we are satisfied with the first appropriate sequence and when the finding of all the optimum sequences is not aimed at, or if it would take too much machine time.

b. It continues the search, but it compares the filter eventually increased by the user with the total processing time of the sequence found and if this latter is the smaller, it uses this as a filter in the further search, ensuring thereby the convergency of the solution. In searching for optimum sequences it is a very important function, when, in order to exclude the possible (and unknown) error in the estimation of the optimum, the value of the filter is increased by a certain "allowance" for decreasing the running time. Corresponding to the program, the search can be interrupted at any time and place and started again with new parameters. With the modification of parameters, e.g. with providing a new filter, the search starts again anew.

c. It proceeds with the search, but lets the value of filter unchanged.

This is useful, when an acceptably short total construction time is only one aspect in choosing the most favourable sequence and it is given only by its upper limit. Correspondingly, namely, to the above, during fitting, the program follows the sum of interruptions and also the spreading of "own" schedules. Thus it becomes possible that the set of appropriate or alternate optimums can be further studied (at present, two such aspects are built into the program: the minimum for the sum of interruptions and the minimum for the spreading of "own" schedules), and finally the program can choose the

most favourable sequence helping thereby the user. The sequences judged as optimum or appropriate on the basis of the total construction time can be printed out.

As a result, the program provides the sequence (sequences) judged as optimum or appropriate by using the filter, together with total construction time and the sum of interruptions or spreading of the "own" schedules belonging to them. However, these two latter are only informative values, since as it follows from their correlation, they can only be decreased at the expense of each other.

Schedules chosen or corresponding to the optimum sequence can be processed afterwards either by the aid of a computer, or manually.

#### 4. Application: Sequencing of more jobs on more machines

For illustrating the above procedure, let us take the following, simplified example:

3 buildings have to be built with 4 work processes on each. Following requirements should be satisfied for the process of construction:

- the same work process is carried out by the same production unit on each building,
- the production unit should work continuously, without interruption,
- the technological sequence is given.

The necessary periods of construction given by work processes and the follow-up times prescribed between work processes are shown in Table 1.

Table 1

Building (i)	Work process (j)				
		1	2	3	4
1	$d_{1,j}$	3	4	5	3
	$z_{1,j}$		2	1	1
2	$d_{2,j}$	4	2	2	1
	$z_{2,j}$		1	2	1
3	$d_{3,j}$	2	1	1	3
	$z_{3,j}$		3	3	4

The problem can be better visualized in a cyclogram representation, the so-called "own" cyclograms representing the separate construction of the individual buildings with the given data and conditions can be drawn (Fig. 2).

Let us examine, how the total construction time changes when changing the sequence of the construction of the individual buildings.

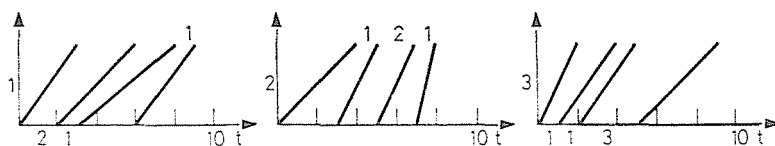


Fig. 2.

Due to our requirements concerning the process of production, when combining the "own" cyclograms, the individual schedule lines are allowed to move only to the right (into positive direction) relative to the zero line and

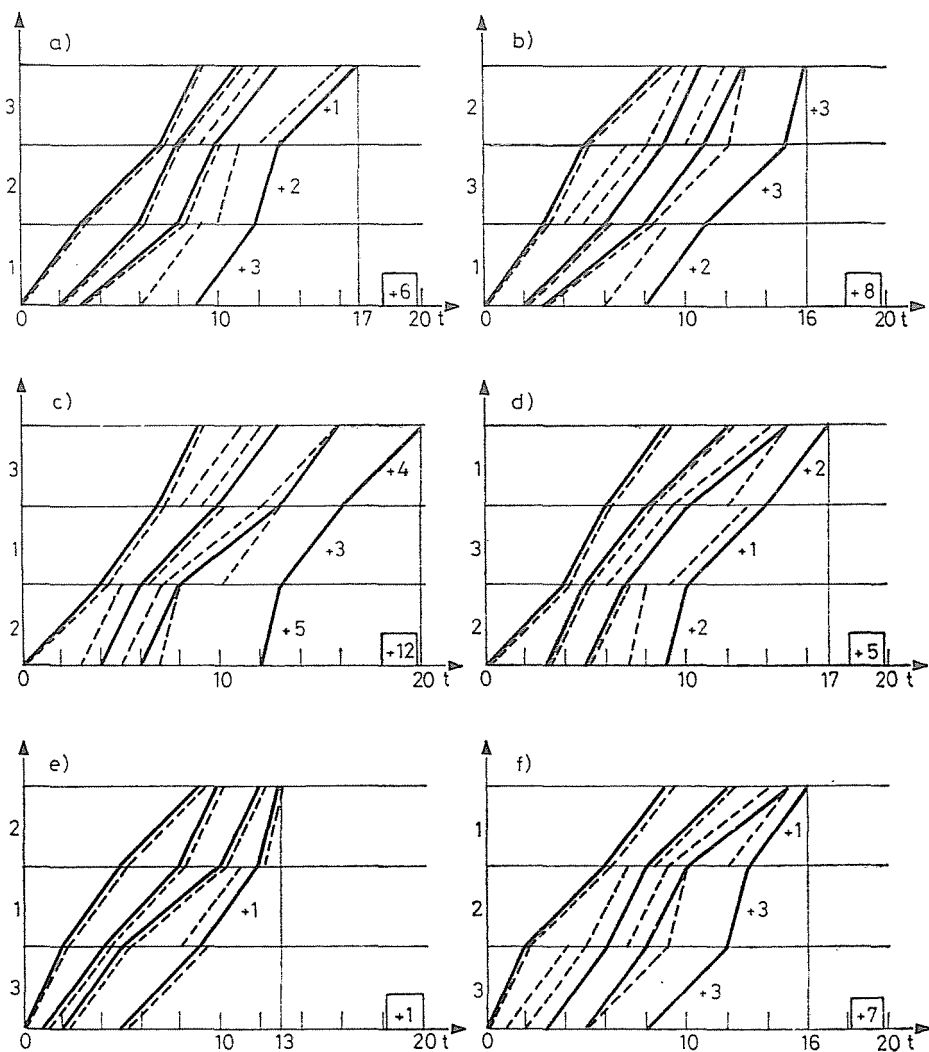


Fig. 3. a. b. c. d. e. f.

to each other. Let us prepare the cyclograms corresponding to the different sequences ( $3! = 6$  pieces of them) (Fig. 3.).

The figures show well the effect of sequence on the total construction time. Even in this simple task, a difference of 100  $(20-13)/13 = 53.8\%$  is found in the total construction time for the optimum and the least favourable sequence. It can also be observed that during fitting into one cyclogram, the individual "own" cyclograms had to be spread to different extents. These spreads are represented by the dotted lines and by the numbers written to the corresponding cyclograms. It is not necessary that the summarized value of the spreads of the "own" cyclograms should be the minimum in the sequence corresponding to the minimum in total construction time (the small numbers in the lower corner of cyclograms).

It should be noted that it is not at all sure that for the optimum solution the same sequences will be obtained if the continuous job of production units is not required. The fact that in our case the optimum sequence would not change with the omission of this strict requirement is only due to the demonstrative nature of the example and to the fact that only 4 work processes are involved for each product. Let us now follow the course of the solution on the enumeration tree (Fig. 4).

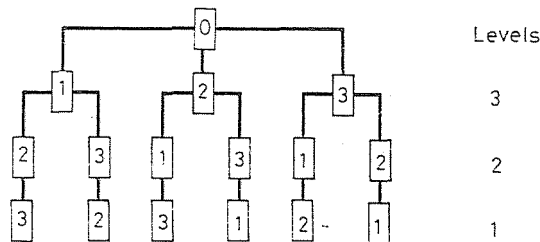


Fig. 4

The levels represent the serial number of a work in a certain sequence. Sequencing is started at the last element. The "0" node (root) is symbolic. Here the first approximation is substituted as filter.

In the first step let us determine the first approximating value, i.e. the theoretically possible minimum of the total construction time. From the summarized work processes the fictive schedule can be determined and by fitting this to its own follow-up vectors corresponding to its "own" schedule and considering also the continuity requirements we choose the shortest construction time from among the possible  $n(n-1)$  ( $3 \times 2$ ) variants. This is given as first approximation, i.e. as first filter. (This minimum can be justified as a theoretically possible lower limit, but it is not at all sure that this is, at the same time, the optimum value as well.)

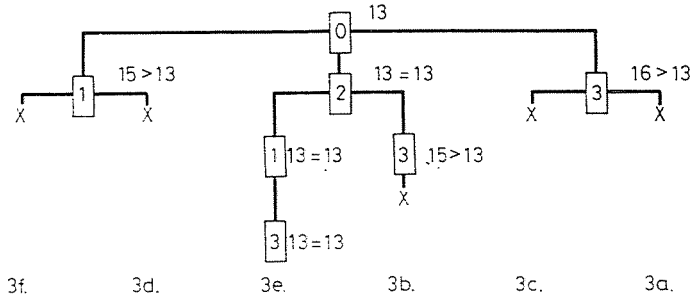


Fig. 5

The examination is carried out on the enumeration tree from the left to the right. In Fig. 5, the values found are shown beside the corresponding node. The nodes at which no numbers stand were not necessary to be examined. Numbers and letters on the bottom of the figure refer to the corresponding cyclograms.

It can be traced on the figure that the suitability of individual sequences is decided at different levels. It turned out already at level 3 that jobs 1 and 3 cannot be the last one, whereas that job 3 cannot be even second turned out only at level 2. Thus  $100(4/6) = 66.6\%$  of the nodes not participating in the optimum sequences could be excluded from the study at level 3, whereas  $100(1/2) = 50\%$  at level 2.

By observing Fig. 5 a little better it can be seen that if 16 or a larger number were taken as first approximation, the program would have provided sequence 3-2-1 as well as a sequence corresponding to the filter value, with 16 days as a target value of total construction time. In the case of a filter smaller than 13, it would turn out already at level 3 that a sequence of such a construction time does not exist. It should be noted that in a larger task, on approaching optimum from below, it would turn out at lower and lower levels that the supposed optimum does not exist. This is why a more accurate first filter value is needed.

If as a subordinate aspect in the study, the minimization of the spread of "own" schedules were also prescribed, then a difference of  $100(12-1)/1 = 1100\%$  (!) would be observed between the target values of the target function. In the given case, this may have an importance in the protection of substance, inner storage, area occupation, capacity calculations etc.

Dr. Tamás TÖRÖK }  
Zoltán VATTAI } H-1521, Budapest